UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/782,732	02/19/2004	Sohail Baig Mohammed	MS1-1849US	3406

22801 7590 06/09/2009

LEE & HAYES, PLLC
601 W. RIVERSIDE AVENUE
SUITE 1400
SPOKANE, WA 99201

EXAMINER	
TAN, ALVIN H	

ART UNIT	PAPER NUMBER
2173	

MAIL DATE	DELIVERY MODE
06/09/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/782,732	MOHAMMED ET AL.	
	Examiner	Art Unit	
	ALVIN H. TAN	2173	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 March 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,4,7-9,11-15,17,19,20,23-26,29-38,40 and 41* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,4,7-9,11-15,17,19,20,23-26,29-38,40 and 41* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *1/21/09, 3/26/09*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Remarks

1. This Office action is responsive to the Request for Continued Examination (RCE) filed under 37 CFR §1.53(d) for the instant application on 3/26/09. Applicants have properly set forth the RCE, which has been entered into the application, and an examination on the merits follows herewith.

2. Claims 1, 4, 7-9, 11-15, 17, 19, 20, 23-26, 29-38, 40, and 41 have been examined and rejected. This Office action is responsive to the amendment filed on 3/26/09, which has been entered in the above identified application.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1, 7, 8, 11-14, 17, 19, 20, 24-26, 29, 31-34, 36-38 and 41 are rejected under 35 U.S.C. 102(e) as being anticipated by Lin et al. (U.S. Patent No. 6,369,835 B1).

Claims 1, 7, 8, 11 (Method)

4-1. Regarding claim 1, Lin teaches a method and computer-readable media comprising examining a plurality of nodes within a media timeline, by disclosing transforming slides for a slide show presentation into movie data and saving the movie data in a movie file for display as a movie by any program capable of playing the movie file *[column 1, lines 6-12]*. Movie data comprises slide video samples, transition video samples, effect samples, and multiple types of media such as video, sound, and sprite in which references to them are disposed on a video track contained in a video media *[column 3, lines 11-37]*.

Lin teaches wherein the media timeline is for exposure over an application programming interface (API), by disclosing that a movie application programming interface may be selected to save the movie data in the movie file *[column 3, lines 3-10]*.

Lin teaches one or more nodes reference respective media and dividing the media timeline into one or more presentations, wherein each presentation describes a rendering of the media for a particular interval of time, by disclosing that the video track references a segment on the track occupied by a sample to a corresponding order and timing of the corresponding objects in the slide show presentation. Each track is a

timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Thus, each media container represents a presentation of a particular media.

Lin teaches wherein each presentation describes a collection of software components that, when executed, provides the described rendering of the media for the particular interval of time, by disclosing that each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Software components include transitions and effects *[column 14, lines 44-55]*, audio data such as background audio *[column 12, lines 53-65]*, voice narration *[column 13, lines 3-17]*, and sound effects *[column 13, lines 18-37]*, and video clips from other applications *[column 18, lines 20-24]*.

Lin teaches wherein the collection of software components include a transform, by disclosing that the movie API provides editing the movie data such as producing/calling up a video effect including fade, wipe, move, swivel, and spiral as well as selecting the number of colors, size, and position *[column 3, lines 38-50]*. Lin teaches wherein the collection of software components comprise at least one of a timeline source, a media source, a media session, a media engine, a source resolver, and a media sink, by disclosing using a track data structure that references the segment on the track occupied by a sample to the corresponding order and timing of the corresponding object in the presentation. Each track is a timeline with marks that denote when the movie API should begin playing samples from an identified media *[column 10,*

lines 43-54]. Additionally, software components may include transitions and effects *[column 14, lines 44-55]*, audio data such as background audio *[column 12, lines 53-65]*, voice narration *[column 13, lines 3-17]*, and sound effects *[column 13, lines 18-37]*, and video clips from other applications *[column 18, lines 20-24]*.

Lin teaches configuring at least one node for communication of events to another node such that a change may be made to the media timeline while the media timeline is rendered, wherein the rendered media timeline is presented on an output device, by disclosing that while the slide show is being rendered to generate the movie file as described in *[figures 3A, 3B, 6-9]*, the user can edit the audio and video samples and tracks *[column 13, lines 50-55; figure 3B, '114', '116']*. A thumbnail movie preview box may be provided so that the user can immediately see what effect a selected change to a movie parameter will have on the play back of the movie before saving the changed parameter *[column 9, lines 15-20]*. The interface showing the preview is displayed on an output device *[column 6, lines 49-50; figure 1]*.

Lin teaches further comprising loading each software component described by a first collection, executing each software component described by the first collection, and loading each software component described by a second collection, wherein each software component that is described by the second collection is loaded during the executing of the first collection, by disclosing that each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Thus, each of the software components would be loaded and executed based on a particular time interval. As

shown in *[figure 11]*, an audio media may be referenced to a primary video track *[column 14, lines 56-65]*. Thus, while the primary video is playing, the various audio samples will be loaded based on their references to the primary video track. Additionally as described in *[column 17, lines 43-48; figure 15B]*, various tracks may be references to the primary video track. Since the movie may be played in a streaming format *[column 9, lines 35-36]*, the objects saved in a particular media will only be loaded and played based on when they are referenced in the timeline.

4-2. Regarding claim 7, Lin discloses method as described in claim 1, further comprising receiving a request from the application over the API to render the media timeline *[column 10, lines 48-54]*.

4-3. Regarding claim 8, Lin discloses method and source as described in claim 1, wherein at least one node is configured to reference an effect to be applied to an output of the media referenced by the node *[column 10, lines 55-65]*.

4-4. Regarding claim 11, Lin discloses one or more computer readable storage media comprising computer executable instruction that, when executed on a computer, direct the computer to perform the method of claims 1 and 12 *[column 5, lines 48-54]*.

Claims 12-14, 17, 19 (Method)

4-5. Regarding claim 12, Lin discloses a method comprising receiving a call from an application over an API for rendering a media timeline wherein the media timeline includes a plurality of nodes, by disclosing transforming slides for a slide show presentation into movie data and saving the movie data in a movie file for display as a movie by any program capable of playing the movie file *[column 1, lines 6-12]*. Movie data comprises slide video samples, transition video samples, effect samples, and multiple types of media such as video, sound, and sprite in which references to them are disposed on a video track contained in a video media *[column 3, lines 11-37]*. A movie application programming interface may be selected to save the movie data in the movie file *[column 3, lines 3-10; column 10, lines 48-54]*.

Lin teaches wherein the plurality of nodes comprises at least a parent node and a child node and two or more nodes reference respective media, by disclosing *[column 14, line 44 to column 15, line 29; figures 10, 11]*. Additionally, other video clips may be integrated within the primary video track *[column 17, line 43 to column 18, line 24; figures 15B, 16B]*.

Lin teaches the media timeline defines one or more presentations including media and rendering the media timeline to output each presentation to an output device, by disclosing that the video track references a segment on the track occupied by a sample to a corresponding order and timing of the corresponding objects in the slide show presentation. Each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Thus, each media container represents a presentation of a particular

media. A thumbnail movie preview box may be provided so that the user can immediately see what effect a selected change to a movie parameter will have on the play back of the movie before saving the changed parameter *[column 9, lines 15-20]*. The interface showing the preview is displayed on an output device *[column 6, lines 49-50; figure 1]*.

Lin teaches dividing the media timeline into one or more presentations, such that each presentation describes a collection of software components utilized to render media for the particular interval of time, by disclosing that each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Software components include transitions and effects *[column 14, lines 44-55]*, audio data such as background audio *[column 12, lines 53-65]*, voice narration *[column 13, lines 3-17]*, and sound effects *[column 13, lines 18-37]*, and video clips from other applications *[column 18, lines 20-24]*.

Lin teaches wherein the collection of software components include a transform by disclosing that the movie API provides editing the movie data such as producing/calling up a video effect including fade, wipe, move, swivel, and spiral as well as selecting the number of colors, size, and position *[column 3, lines 38-50]*. Lin teaches wherein the collection of software components comprise at least one of a timeline source, a media source, a media session, a media engine, a source resolver, and a media sink, by disclosing using a track data structure that references the segment on the track occupied by a sample to the corresponding order and timing of the

corresponding object in the presentation. Each track is a timeline with marks that denote when the movie API should begin playing samples from an identified media *[column 10, lines 43-54]*. Additionally, software components may include transitions and effects *[column 14, lines 44-55]*, audio data such as background audio *[column 12, lines 53-65]*, voice narration *[column 13, lines 3-17]*, and sound effects *[column 13, lines 18-37]*, and video clips from other applications *[column 18, lines 20-24]*.

Lin teaches configuring at least one node for communication of events to another node such that a change may be made to the media timeline while the media timeline is rendered by performing at least one of changing to a property of the at least one node, adding one or more additional nodes as a child to the at least one node, removing one or more nodes that are children of the at least one node, adding an effect to the at least one node, and removing an effect from the at least one node, by disclosing that while the slide show is being rendered to generate the movie file as described in *[figures 3A, 3B, 6-9]*, the user can edit the audio and video samples and tracks *[column 13, lines 50-55; figure 3B, '114', '116']*.

Lin teaches loading each software component described by a first collection, executing each software component described by the first collection, and loading each software component described by a second collection, by disclosing that each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Thus, each of the software components would be loaded and executed based on a particular time interval.

4-6. Regarding claim 13, Lin discloses method as described in claim 12, wherein the rendering further comprises examining the media timeline *[column 19, lines 46-49]*.

4-7. Regarding claim 14, Lin discloses method as described in claim 12, wherein each collection does not change for the particular interval of time described by a respective presentation *[column 8, lines 11-20]*.

4-8. Regarding claim 17, Lin discloses method and source as described in claim 12, wherein at least one node is configured to reference an effect to be applied to an output of the media referenced by the node *[column 10, lines 55-65]*.

4-9. Regarding claim 19, Lin discloses one or more computer readable storage media comprising computer executable instruction that, when executed on a computer, direct the computer to perform the method of claims 1 and 12 *[column 5, lines 48-54]*.

Claims 20, 24, 25 (Computer Readable Media)

4-10. Regarding claim 20, Lin teaches the claim wherein a media timeline is for exposure via an API to one or more applications, the media timeline includes a plurality of nodes, by disclosing transforming slides for a slide show presentation into movie data and saving the movie data in a movie file for display as a movie by any program capable of playing the movie file *[column 1, lines 6-12]*. Movie data comprises slide video samples, transition video samples, effect samples, and multiple types of media

such as video, sound, and sprite in which references to them are disposed on a video track contained in a video media *[column 3, lines 11-37]*. A movie application programming interface may be selected to save the movie data in the movie file *[column 3, lines 3-10; column 10, lines 48-54]*.

Lin teaches wherein the plurality of nodes comprises at least a parent node and a child node and at least two nodes reference respective media, by disclosing *[column 14, line 44 to column 15, line 29; figures 10, 11]*.

Lin teaches each presentation describes rendering of respective media to an output device for a particular interval of time, by disclosing that the video track references a segment on the track occupied by a sample to a corresponding order and timing of the corresponding objects in the slide show presentation. Each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Thus, each media container represents a presentation of a particular media. A thumbnail movie preview box may be provided so that the user can immediately see what effect a selected change to a movie parameter will have on the play back of the movie before saving the changed parameter *[column 9, lines 15-20]*. The interface showing the preview is displayed on an output device *[column 6, lines 49-50; figure 1]*.

Lin teaches wherein each presentation describes a collection of software components that, when executed, provides the described rendering of media for the particular interval of time, by disclosing that each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an

identified media container *[column 10, lines 43-54]*. Software components include transitions and effects *[column 14, lines 44-55]*, audio data such as background audio *[column 12, lines 53-65]*, voice narration *[column 13, lines 3-17]*, and sound effects *[column 13, lines 18-37]*, and video clips from other applications *[column 18, lines 20-24]*.

Lin teaches wherein the collection of software components include a transform, by disclosing that the movie API provides editing the movie data such as producing/calling up a video effect including fade, wipe, move, swivel, and spiral as well as selecting the number of colors, size, and position *[column 3, lines 38-50]*. Lin teaches wherein the collection of software components comprise at least one of a timeline source, a media source, a media session, a media engine, a source resolver, and a media sink, by disclosing using a track data structure that references the segment on the track occupied by a sample to the corresponding order and timing of the corresponding object in the presentation. Each track is a timeline with marks that denote when the movie API should begin playing samples from an identified media *[column 10, lines 43-54]*. Additionally, software components may include transitions and effects *[column 14, lines 44-55]*, audio data such as background audio *[column 12, lines 53-65]*, voice narration *[column 13, lines 3-17]*, and sound effects *[column 13, lines 18-37]*, and video clips from other applications *[column 18, lines 20-24]*.

Lin teaches computer executable instructions that, when executed on the computer, direct the computer to load each software component described by a first collection, execute each software component described by the first collection, and load

each software component described by a second collection, by disclosing that each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Thus, each of the software components would be loaded and executed based on a particular time interval.

4-11. Regarding claim 24, Lin discloses method and source as described in claim 20, wherein at least one node is configured to reference an effect to be applied to an output of said media referenced by the node *[column 10, lines 55-65]*.

4-12. Regarding claim 25, Lin discloses method and source as described in claim 20, wherein at least one node is configured for communication of events to another node such that a change may be made to the media timeline while the media timeline is rendered, by disclosing that while the slide show is being rendered to generate the movie file as described in *[figures 3A, 3B, 6-9]*, the user can edit the audio and video samples and tracks *[column 13, lines 50-55; figure 3B, '114', '116']*.

Claims 26, 29, 31, 32 (System)

4-13. Regarding claim 26, Lin teaches a system comprising a memory, a processor coupled to the memory, a plurality of media, and plurality of applications, by disclosing a computer system *[figure 1]* for transforming slides for a slide show presentation into movie data and saving the movie data in a movie file for display as a movie by any

program capable of playing the movie file *[column 1, lines 6-12]*. Movie data comprises slide video samples, transition video samples, effect samples, and multiple types of media such as video, sound, and sprite in which references to them are disposed on a video track contained in a video media *[column 3, lines 11-37]*.

Lin teaches an infrastructure layer that provides an API for the plurality of applications which exposes a media timeline that describes one or more presentations of the plurality of media, by disclosing that a movie application programming interface may be selected to save the movie data in the movie file *[column 3, lines 3-10]*.

Lin teaches managing rendering of the one or more presentations, wherein each presentation describes rendering of media to an output device for a particular interval of time, by disclosing that the video track references a segment on the track occupied by a sample to a corresponding order and timing of the corresponding objects in the slide show presentation. Each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Thus, each media container represents a presentation of a particular media. A thumbnail movie preview box may be provided so that the user can immediately see what effect a selected change to a movie parameter will have on the play back of the movie before saving the changed parameter *[column 9, lines 15-20]*. The interface showing the preview is displayed on an output device *[column 6, lines 49-50; figure 1]*.

Lin teaches wherein each presentation describes a collection of software components configured for dynamic loading such that the collection of software

components provide the described rendering of the media for the particular interval of time, by disclosing that each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Software components include transitions and effects *[column 14, lines 44-55]*, audio data such as background audio *[column 12, lines 53-65]*, voice narration *[column 13, lines 3-17]*, and sound effects *[column 13, lines 18-37]*, and video clips from other applications *[column 18, lines 20-24]*. Various software components such as transitions and audio tracks may be loaded based on options selected by the user while the slide show is being rendered to generate the movie file as described in *[figures 3A, 3B, 6-9]*. The user can edit the audio and video samples and tracks *[column 13, lines 50-55; figure 3B, '114', '116']*.

Lin teaches wherein the collection of software components include a transform by disclosing that the movie API provides editing the movie data such as producing/calling up a video effect including fade, wipe, move, swivel, and spiral as well as selecting the number of colors, size, and position *[column 3, lines 38-50]*. Lin teaches wherein the collection of software components comprise at least one of a timeline source, a media source, a media session, a media engine, a source resolver, and a media sink, wherein the collection of software components are loaded only when needed, by disclosing using a track data structure that references the segment on the track occupied by a sample to the corresponding order and timing of the corresponding object in the presentation. Each track is a timeline with marks that denote when the movie API should begin playing samples from an identified media *[column 10, lines 43-*

54]. The software components making up the movie file will be loaded when the user plays the movie file either initially or in a streaming format *[column 17, lines 6-32; column 9, lines 35-36]*.

4-14. Regarding claim 29, Lin discloses system as described in claim 26, wherein the collection does not change for the particular interval of time described *[column 8, lines 11-20]*.

4-15. Regarding claim 31, Lin discloses the system as described in claim 26, wherein the media timeline includes a plurality of nodes and at least two nodes reference respective media, by disclosing movie data comprises slide video samples, transition video samples, effect samples, and multiple types of media such as video, sound, and sprite in which references to them are disposed on a video track contained in a video media *[column 3, lines 11-37]*. Thus, the references to the movie data represent the plurality of nodes.

Lin teaches at least one said node is configured to reference an effect to be applied to an output of media referenced by the node *[column 10, lines 55-65]*.

4-16. Regarding claim 32, Lin discloses the system as described in claim 26, wherein the media timeline includes a plurality of nodes and at least two nodes reference respective media, by disclosing movie data comprises slide video samples, transition video samples, effect samples, and multiple types of media such as video, sound, and

sprite in which references to them are disposed on a video track contained in a video media *[column 3, lines 11-37]*.

Lin teaches wherein at least one node is configured for communication of events to another node such that a change may be made to the media timeline while the media timeline is rendered *[column 19, lines 46-52]*.

Claims 33, 34, 36-38, 41 (Timeline Source)

4-17. Regarding claim 33, Lin teaches means for dividing a media timeline into one or more presentations each describing a rendering of one or more media during a particular interval of time, by disclosing transforming slides for a slide show presentation into movie data and saving the movie data in a movie file for display as a movie by any program capable of playing the movie file *[column 1, lines 6-12]*. A video track references a segment on the track occupied by a sample to a corresponding order and timing of the corresponding objects in the slide show presentation. Each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Thus, each media container represents a presentation of a particular media.

Lin teaches wherein the media timeline exposes a plurality of nodes to a plurality of applications, wherein one or more nodes reference respective said media, by disclosing that movie data comprises slide video samples, transition video samples, effect samples, and multiple types of media such as video, sound, and sprite in which

references to them are disposed on a video track contained in a video media *[column 3, lines 11-37]*.

Lin teaches wherein the media timeline is configured for dynamic loading such that metadata included in at least one node specifies a collection of nodes to be loaded when the media timeline is rendered, wherein the rendered media timeline is presented on an output device, by disclosing that while the slide show is being rendered to generate the movie file as described in *[figures 3A, 3B, 6-9]*, the user can edit the audio and video samples and tracks *[column 13, lines 50-55; figure 3B, '114', '116']*. A thumbnail movie preview box may be provided so that the user can immediately see what effect a selected change to a movie parameter will have on the play back of the movie before saving the changed parameter *[column 9, lines 15-20]*. The interface showing the preview is displayed on an output device *[column 6, lines 49-50; figure 1]*.

Lin teaches means for determining a topology for each presentation, wherein the topology references a collection of software components that, when executed, provides the rendering, by disclosing software components such as transitions and effects *[column 14, lines 44-55]*, audio data such as background audio *[column 12, lines 53-65]*, voice narration *[column 13, lines 3-17]*, and sound effects *[column 13, lines 18-37]*, and video clips from other applications *[column 18, lines 20-24]* that make up the presentation as a whole.

Lin teaches media processor means for executing the topology for each presentation that is described by the timeline, by disclosing the playback of movies *[figures 13, 15A,B, 16A,B]*.

4-18. Regarding claim 34, Lin discloses timeline source as described in claim 33, wherein each collection does not change for the particular interval of time described by a respective presentation *[column 8, lines 11-20]*.

4-19. Regarding claim 36, Lin discloses method and source as described in claim 33, wherein at least one node is configured to reference an effect to be applied to an output of media referenced by the node *[column 10, lines 55-65]*.

4-20. Regarding claim 37, Lin discloses method and source as described in claims 33, wherein at least one node is configured for communication of events to another node such that a change may be made to a property of the at least one node while the media timeline is rendered *[column 19, lines 46-52]*.

4-21. Regarding claim 38, Lin discloses timeline source as described in claim 33, wherein the media timeline is configured for dynamic creation such that at least one node is created while the media timeline is rendered *[column 11, lines 9-16]*.

4-22. Regarding claim 41, Lin discloses timeline source as described in claim 33, further comprising means for translating a time specified by one node for rendering the one node with respect to a time specified by another node *[fig. 2: items 66, 78B, 78C and 98]*.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 4, 9, 15, 23, 30, 35, and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lin et al. (U.S. Patent No. 6,369,835 B1) in view of French et al. (U.S. Patent No. 6,266,053 B1).

Claims 4, 9

Claim 15

Claim 23

Claim 30

Claim 35, 40

6-1. Regarding claims 4, 15, 23, 30, and 35, Lin teaches the claims as described in claims 1, 12, 20, 26, and 33, respectively. Although Lin teaches, "dividing the media timeline into the one or more presentations" *[column 10, lines 43-48]*, Lin does not expressly teach "each presentation describes a respective partial topology of software components; and the respective partial topology is for resolving into a full topology that references each software component utilized to provide a respective presentation."

French does teach topology *[column 10, lines 17-24: graph]*. Therefore, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to include partial or full topology in Lin's systems. One would have been motivated to do so in order to efficiently utilized topology on presentation *[column 9, lines 24-30: graph]*.

6-2. Regarding claims 9 and 40, Lin teaches the claims as described in claims 1 and 33, respectively. Lin does not expressly teach "wherein at least one node is specified as read-only". French discloses a similar method of using a timeline with nodes to represent a scene *[column 3, lines 47-64]*. Read-only reference to input objects are used by the operator *[column 9, lines 24-30]*. It would have been obvious to one having ordinary skill in the art at the time of the invention was made to include read-only in Lin's systems. One would have been motivated to do so in order to efficiently protect nodes *[column 9, lines 24-30: object]* from being overridden.

Response to Arguments

7. The Examiner acknowledges the Applicant's amendments to claims 1, 4, 8, 9, 11, 12, 14, 15, 17, 19, 20, 23-26, 30-38, 41, and 41 and the cancellation of claim 6. Regarding independent claim 1, the Applicant alleges that Lin et al. (U.S. Patent No. 6,369,835 B1), as described in the previous Office action, does not explicitly teach, "configuring at least one said node for communication of events to another said node such that a change may be made to the media timeline while the media timeline is

rendered, wherein the rendered media timeline is presented on an output device", as has been amended to the claim. Contrary to Applicant's arguments, Lin discloses that while the slide show is being rendered to generate the movie file as described in *[figures 3A, 3B, 6-9]*, the user can edit the audio and video samples and tracks *[column 13, lines 50-55; figure 3B, '114', '116']*. A thumbnail movie preview box may be provided so that the user can immediately see what effect a selected change to a movie parameter will have on the play back of the movie before saving the changed parameter *[column 9, lines 15-20]*. The interface showing the preview is displayed on an output device *[column 6, lines 49-50; figure 1]*.

Applicant alleges that Lin does not explicitly teach, "loading each software component described by a second collection, wherein each software component that is described by the second collection is loaded during the executing of the first collection," as has been amended to the claim. Contrary to Applicant's arguments, Lin discloses that each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Since the movie may be played in a streaming format *[column 9, lines 35-36],* each of the software components would be loaded and executed based on a particular time interval. As shown in *[figure 11]*, an audio media may be referenced to a primary video track *[column 14, lines 56-65]*. Thus, while the primary video is playing, the various audio samples will be loaded based on their references to the primary video track. Additionally as described in *[column 17, lines 43-48; figure 15B]*, various tracks may be references to the primary video track.

Regarding independent claim 12, Applicant alleges that Lin does not explicitly teach, "rendering the media timeline to output each presentation to an output device", as has been amended to the claim. Contrary to Applicant's arguments, Lin discloses that the video track references a segment on the track occupied by a sample to a corresponding order and timing of the corresponding objects in the slide show presentation. Each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Thus, each media container represents a presentation of a particular media. A thumbnail movie preview box may be provided so that the user can immediately see what effect a selected change to a movie parameter will have on the play back of the movie before saving the changed parameter *[column 9, lines 15-20]*. The interface showing the preview is displayed on an output device *[column 6, lines 49-50; figure 1]*.

Applicant alleges that Lin does not explicitly teach, "loading each software component described by a first collection, executing each software component described by the first collection, and loading each software component described by a second collection". Contrary to Applicant's arguments, Lin discloses that each track is a timeline with marks (references) that denote when the movie API should begin playing samples from an identified media container *[column 10, lines 43-54]*. Thus, each of the software components would be loaded and executed based on a particular time interval.

Further regarding independent claim 12, Applicant alleges that Lin, as described in the previous Office action, does not explicitly teach, "configuring at least one said

node for communication of events to another said node such that a change may be made to the media timeline while the media timeline is rendered by performing at least one of the following: changing to a property of the at least one said node; adding one or more additional said nodes as a child to the at least one said node; removing one or more said nodes that are children of the at least one said node; adding an effect to the at least one said node; and removing an effect from the at least one said node, " as has been amended to the claim. Contrary to Applicant's arguments, Lin discloses that a primary video track is made up of video effects and various audio tracks that reference the primary video track *[column 14, line 44 to column 15, line 29; figures 10, 11]*. Additionally, other video clips may be integrated within the primary video track *[column 17, line 43 to column 18, line 24; figures 15B, 16B]*. These transitions and effects *[column 14, lines 44-55]*, audio data such as background audio *[column 12, lines 53-65]*, voice narration *[column 13, lines 3-17]*, sound effects *[column 13, lines 18-37]*, and video clips from other applications *[column 18, lines 20-24]* may be considered nodes. Lin discloses that while the slide show is being rendered to generate the movie file as described in *[figures 3A, 3B, 6-9]*, the user can edit the audio and video samples and tracks *[column 13, lines 50-55; figure 3B, '114', '116']*.

Similar arguments have been presented for claims 20, 26, and 33 and thus, Applicant's arguments are not persuasive for the same reasons.

Regarding claims 20, 26, and 33, Applicant alleges that Lin, as described in the previous Office action, does not explicitly teach dynamic loading. Examiner notes that dynamic loading may be interpreted as any loading done on the fly. Contrary to

Applicant's arguments Lin discloses that various software components such as transitions and audio tracks may be loaded based on options selected by the user while the slide show is being rendered to generate the movie file as described in *[figures 3A, 3B, 6-9]*. The user can edit the audio and video samples and tracks *[column 13, lines 50-55; figure 3B, '114', '116']*. The movie may also be saved in a streaming format *[column 9, lines 35-36]*.

Applicant states that dependent claims 4, 6, 7-9, 11, 13-15, 17, 19, 23-25, 29-32, 34-38, 40, and 41 recite all the limitations of the independent claims, and thus, are allowable in view of the remarks set forth regarding independent claims 1, 12, 20, 26, and 33. However, as discussed above, Lin is considered to teach claims 1, 12, 20, 26, and 33, and consequently, claims 4, 6, 7-9, 11, 13-15, 17, 19, 23-25, 29-32, 34-38, 40, and 41 are rejected.

Conclusion

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to ALVIN H. TAN whose telephone number is (571)272-8595. The examiner can normally be reached on Mon-Fri 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kieu Vu can be reached on 571-272-4057. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Alvin H Tan/
Examiner, Art Unit 2173